In the claims:

For the Examiner's convenience, all pending claims are presented below with changes shown in accordance with the mandatory amendment format.

1. (Currently Amended) A computer-implemented method comprising:

distributing ~~a computer program component,~~ a device driver to execute functionality under command from a kernel, wherein the device driver ~~which~~ includes code defining functionality associated with the device driver ~~a computer program module~~ and excludes a header, wherein the header includes unique symbols associated with the kernel and version identification data of the kernel~~, for the computer program module to execute the functionality under command from a master computer program~~; and

distributing ~~an installation module~~ the device driver which, when run on a computer, dynamically creates the header information for the device driver by obtaining ~~obtains~~ the version identification data and the associated unique symbols from the kernel ~~the master computer program and combines the version identification data and the computer program component to define the computer program module~~.

2. (Currently Amended) The computer-implemented method of Claim 1, in which the kernel ~~master computer program~~ is part of an operating system ~~and the computer program module is a device driver~~, wherein the kernel is ~~master computer program~~ ~~being~~ identifiable by the version identification data.

3. (Previously Presented) The computer-implemented method of Claim 2, in which the operating system is selected from the group including a Linux operating system and a UNIX operating system.

4. (Currently Amended) The computer-implemented method of Claim 3, in which ~~the functionality included in~~ the device driver executes ~~computer program component allows the computer program module to execute~~ an application program interface (API) exported from the kernel ~~master computer program~~.

5. (Currently Amended) The computer-implemented method of Claim 3, further comprising compiling the device driver ~~computer program component~~ into an object file prior to distribution ~~of the computer program module~~.

6. (Currently Amended) The computer-implemented method of Claim 5, further comprising obtaining the version identification data from the operating system and generating a version object file that includes the version identification data.

7. (Currently Amended) The computer-implemented method of Claim 6, further comprising linking the version object file and the device driver ~~computer program component~~.

8. (Currently Amended) The computer-implemented method of Claim 7, further comprising obtaining a kernel specific address of a module list and passing the address to the driver ~~computer program module~~.

9. (Currently Amended) The computer-implemented method of Claim 2, in which the device driver is one of a printer driver, a serial port device driver, an ethernet ~~ethernet~~ device driver, and a disk drive device driver.

10. (Cancelled)

11. (Currently Amended) A computer program product including a medium readable by a computer, the medium carrying instructions which, when executed by the computer, cause the computer to:

distribute a device driver to execute functionality under command from a kernel, wherein the device driver includes code defining functionality associated with the device driver and excludes a header, wherein the header includes unique symbols associated with the kernel and version identification data of the kernel; and

distribute the device driver which, when run on a computer, dynamically creates the header information for the device driver by obtaining the version identification data and the associated unique symbols from the kernel ~~identify a computer program component which includes object code defining functionality associated with the product and excludes version identification data for the product to execute the functionality under command from a master computer program; obtain the version identification data from the master computer program and combine the version identification data and the computer program component to define a computer program module; and store the computer program module in memory~~.

12. (Currently Amended) The product of Claim 11, in which the kernel ~~master computer program~~ is part of an operating system ~~and the computer program module is a device driver~~, wherein the kernel ~~master computer program~~ being identifiable by the version identification data.

13. (Currently Amended) The product of Claim 12, in which the kernel ~~master computer program~~ is selected from the group including a Linux operating system and a UNIX operating system.

14. (Currently Amended) The product of Claim 13, in which the functionality included in the driver executes ~~computer program component allows the computer program module to execute~~ at least one application program interface (API) exported from the kernel ~~master computer program~~.

15. (Currently Amended) The product of Claim 14, further comprising obtaining the version identification data from the operating system and generating a version object file that includes the version identification data.

16. (Cancelled)

17. (Currently Amended) The product of Claim 15 ~~16~~, further comprising obtaining a kernel specific address of a module list and passing the address to the driver ~~computer program product~~.

18. (Currently Amended) The product of Claim 17, in which the device driver ~~computer program product~~ retrieves a module list export head and imports the required application program interfaces (APIs) ignoring the version identification data.

19. (Previously Presented) The product of Claim 13, in which the device driver is dynamically loaded in a Linux kernel.

20. (Currently Amended) The product of Claim 11, in which an installation module forms part of the device driver ~~computer program component~~.

21-25. (Cancelled)